Configuring Seamless Integration of EVPN with L3VPN (MPLS SR)

This chapter contains the following sections:

- Information About Configuring Seamless Integration of EVPN with L3VPN (MPLS SR), on page 1
- Guidelines and Limitations for Configuring Seamless Integration of EVPN with L3VPN (MPLS SR), on page 3
- Configuring Seamless Integration of EVPN with L3VPN (MPLS SR), on page 4
- Example Configuration for Configuring Seamless Integration of EVPN with L3VPN (MPLS SR), on page 8

Information About Configuring Seamless Integration of EVPN with L3VPN (MPLS SR)

Data Center (DC) deployments have adopted VXLAN EVPN for its benefits such as EVPN control-plane learning, multitenancy, seamless mobility, redundancy, and easier POD additions. Similarly, the CORE is either an Label Distribution Protocol (LDP)-based MPLS L3VPN network or transitioning from the traditional MPLS L3VPN LDP-based underlay to a more sophisticated solution like Segment Routing (SR). Segment Routing is adopted for its benefits such as:

- Unified IGP and MPLS control planes
- Simpler traffic engineering methods
- Easier configuration
- SDN adoption

With two different technologies, one within the data center (DC) and one in the CORE, there is a natural necessity to handoff from VXLAN to an MPLS-based core at the DCI nodes, which sit on the edge of the DC domain, interfacing with the Core edge router.
In the previous diagram, two DC pods, each running VXLAN, are being Layer 3 extended over a WAN/Core running MPLS/SR. Another method is classical MPLS L3VPN using LDP. The edge devices in the DC domain (border PE1, PE2, PE3, and PE4) are the DCI nodes doing the handoff between VXLAN and the MPLS-based Core network.
Guidelines and Limitations for Configuring Seamless Integration of EVPN with L3VPN (MPLS SR)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Cisco Nexus 3600</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>VXLAN EVPN to SR-L3VPN</td>
<td>Yes</td>
<td>Extend Layer 3 connectivity between different DC pods Underlay IGP/BGP with SR extensions.</td>
</tr>
<tr>
<td>VXLAN EVPN to SR-L3VPN</td>
<td>Yes</td>
<td>Extend Layer 3 connectivity between DC POD running VXLAN and any domain(DC or CORE) running SR.</td>
</tr>
<tr>
<td>VXLAN EVPN to MPLS L3VPN (LDP)</td>
<td>Yes</td>
<td>Underlay is LDP.</td>
</tr>
</tbody>
</table>

The following Cisco Nexus switches are supported:
- 3600 platform switches

The following features are supported:
- Layer 3 orphans
- Layer 3 hand-off
- 256 peers/nodes within a VXLAN DC domain
- Layer 3 physical interfaces type for Core facing ports
- Per-VRF labels

The following features are not supported:
- vPC for redundancy
- Subnet stretches across the DC domain
- SVI/Sub-interfaces configured MAC addresses
- Statistics for Cisco Nexus 3600 platform switches.
- SVI towards the MPLS core for Cisco Nexus 3600 platform switches.
- End-to-End Time to Live (TTL) support only in pipe mode for handoff scenario.
- End-to-End Explicit Congestion Notification (ECN) for handoff scenario.

Cisco Nexus 3600 platform switches support Segment Routing and LDP.
For Cisco Nexus 3600 platform switches, segment routing and LDP cannot co-exist.
Configuring Seamless Integration of EVPN with L3VPN (MPLS SR)

The following procedure imports and reorigimates the routes from the VXLAN domain to the MPLS domain and in the other direction.

Before you begin

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>feature-set mpls</td>
<td>Enable MPLS feature set.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# feature-set mpls</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>nv overlay evpn</td>
<td>Enable VXLAN.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# nv overlay evpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>feature bgp</td>
<td>Enable BGP.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# feature bgp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td>feature mpls l3vpn</td>
<td>Enable Layer 3 VPN.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# feature mpls l3vpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td></td>
</tr>
<tr>
<td>feature mpls segment-routing</td>
<td>Enable Segment Routing.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# feature mpls segment-routing</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td></td>
</tr>
<tr>
<td>feature interface-vlan</td>
<td>Enable interface VLAN.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# feature interface-vlan</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>feature vn-segment-vlan-based</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)# feature vn-segment-vlan-based</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>feature nv overlay</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)# feature nv overlay</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>router bgp autonomous-system-number</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)# router bgp 1</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>address-family ipv4 unicast</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-router)# address-family ipv4 unicast</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>redistribute direct route-map route-map-name</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-router-af)# redistribute direct route-map passall</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td>network address</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-router-af)# network 0.0.0.0/0</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-router-af)# exit</td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td>address-family l2vpn evpn</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-router)# address-family l2vpn evpn</td>
</tr>
<tr>
<td><strong>Step 17</strong></td>
<td>neighbor address remote-as number</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-router)# neighbor 108.108.108.108 remote-as 65535</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| 18   | **update-source type/id**  
  Example:  
  switch(config-router-af)# update-source loopback100 | Define interface for eBGP peering. |
| 19   | **ebgp-multihop number**  
  Example:  
  switch(config-router)# ebgp-multihop 10 | Specifies multihop TTL for remote peer. The range of `number` is from 2 to 255. |
| 20   | **address-family ipv4 unicast**  
  Example:  
  switch(config-router)# address-family ipv4 unicast | Configure the address family for IPv4. |
| 21   | **send-community extended**  
  Example:  
  switch(config-router-af)# send-community extended | Configures community for BGP neighbors. |
| 22   | **exit**  
  Example:  
  switch(config-router-af)# exit | Exit command mode. |
| 23   | **address-family vpnv4 unicast**  
  Example:  
  switch(config-router)# address-family vpnv4 unicast | Configure the address family for IPv4. |
| 24   | **send-community extended**  
  Example:  
  switch(config-router-af)# send-community extended | Configures community for BGP neighbors. |
| 25   | **import l2vpn evpn reoriginate**  
  Example:  
  switch(config-router)# import l2vpn evpn reoriginate | Reoriginates the route with new RT. Can be extended to use an optional route-map. |
| 26   | **neighbor address remote-as number**  
  Example:  
  switch(config-router)# neighbor 175.175.175.2 remote-as 65535 | Define eBGP neighbor IPv4 address and remote Autonomous-System (AS) number. |
| 27   | **address-family ipv4 unicast**  
  Example:  
<p>| Configure the address family for IPv4. |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch(config-router)# address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 28</strong> send-community extended</td>
<td>Configures community for BGP neighbors.</td>
</tr>
<tr>
<td>Example: switch(config-router-af)# send-community extended</td>
<td></td>
</tr>
<tr>
<td><strong>Step 29</strong> exit</td>
<td>Exit command mode.</td>
</tr>
<tr>
<td>Example: switch(config-router-af)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 30</strong> address-family ipv6 unicast</td>
<td>Configure the IPv6 unicast address family. This is required for IPv6 over VXLAN with an IPv4 underlay.</td>
</tr>
<tr>
<td>Example: switch(config-router)# address-family ipv6 unicast</td>
<td></td>
</tr>
<tr>
<td><strong>Step 31</strong> send-community extended</td>
<td>Configures community for BGP neighbors.</td>
</tr>
<tr>
<td>Example: switch(config-router-af)# send-community extended</td>
<td></td>
</tr>
<tr>
<td><strong>Step 32</strong> exit</td>
<td>Exit command mode.</td>
</tr>
<tr>
<td>Example: switch(config-router-af)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 33</strong> address-family l2vpn evpn</td>
<td>Configure L2VPN address family.</td>
</tr>
<tr>
<td>Example: switch(config-router)# address-family l2vpn evpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 34</strong> send-community extended</td>
<td>Configures community for BGP neighbors.</td>
</tr>
<tr>
<td>Example: switch(config-router-af)# send-community extended</td>
<td></td>
</tr>
<tr>
<td><strong>Step 35</strong> exit</td>
<td>Exit command mode.</td>
</tr>
<tr>
<td>Example: switch(config-router-af)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 36</strong> import vpn unicast reoriginate</td>
<td>Reoriginate the route with new RT. Can be extended to use an optional route-map.</td>
</tr>
<tr>
<td>Example: switch(config-router)# import vpn unicast reoriginate</td>
<td></td>
</tr>
</tbody>
</table>
Example Configuration for Configuring Seamless Integration of EVPN with L3VPN (MPLS SR)

The following is a sample CLI configuration that is required to import and reoriginate the routes from the VXLAN domain to the MPLS domain and in the reverse direction.

```
switch# sh running-config

!Command: show running-config
!Running configuration last done at: Sat Mar 17 10:00:40 2001
!Time: Sat Mar 17 12:50:12 2001

version 9.2(2) Bios:version 05.22
hardware profile multicast max-limit lpm-entries 0

hostname switch
install feature-set mpls
vdc Scrimshaw id 1
  allow feature-set mpls
  limit-resource vlan minimum 16 maximum 4094
  limit-resource vrf minimum 2 maximum 4096
  limit-resource port-channel minimum 0 maximum 511
  limit-resource u4route-mem minimum 248 maximum 248
  limit-resource u6route-mem minimum 96 maximum 96
  limit-resource m4route-mem minimum 96 maximum 96
  limit-resource m6route-mem minimum 90 maximum 90
  limit-resource m6route-mem minimum 8 maximum 8
feature-set mpls

feature telnet
feature bash-shell
feature sftp-server
nv overlay evpn
feature ospf
feature bgp
feature mpls l3vpn
feature mpls segment-routing
feature interface-vlan
feature vn-segment-vlan-based
feature bfd
feature nv overlay

no password strength-check
username admin password 5
$5$eEI.wtRs$txfevWxMj/upb/1dJeXy5rNvFYKymzz3Zmc.fpuTp
  1 role network-admin
ip domain-lookup
ccp profile strict
snmp-server user admin network-admin auth md5 0x116815e4934ab1f854dce5dd673f33d7
  priv 0x116815e4934ab1f854dce5dd673f33d7 localizedkey
rmon event 1 description FATAL(1) owner PMON@FATAL
rmon event 2 description CRITICAL(2) owner PMON@CRITICAL
rmon event 3 description ERROR(3) owner PMON@ERROR
rmon event 4 description WARNING(4) owner PMON@WARNING
rmon event 5 description INFORMATION(5) owner PMON@INFO

mpls label range 30000 40000 static 6000 8000
vlan 1-2,100,200,555
segment-routing mpls
  global-block 30000 40000
vlan 555
```
vn-segment 55500
route-map ALL permit 10
route-map SRmap permit 10
set label-index 666
route-map ULAY_NETWORK permit 10
set label-index 600
route-map passall permit 10
vrf context ch5_swap
  ip route 199.1.1.0/24 16.1.1.2
  ip route 200.1.1.0/24 16.1.1.2
vrf context evpn
  vni 55500
  rd auto
  address-family ipv4 unicast
    route-target import 100:55500
    route-target import 100:55500 evpn
    route-target import 6:6000
    route-target export 100:55500
    route-target export 100:55500 evpn
    route-target export 6:6000
  address-family ipv6 unicast
    route-target import 6:6000
    route-target export 6:6000
vrf context management
  ip route 0.0.0.0/0 172.31.144.1
hardware forwarding unicast trace
vlan configuration 2
  ip igmp snooping static-group 225.1.1.1 interface Ethernet1/9

interface Vlan1
interface Vlan555
  no shutdown
  vrf member evpn

interface nvel
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 55500 associate-vrf

interface Ethernet1/12
  mpls ip forwarding
  no shutdown

interface Ethernet1/13

interface Ethernet1/14
  no shutdown

interface Ethernet1/15
  no shutdown

interface Ethernet1/16
  no shutdown

interface Ethernet1/17
  no shutdown

interface Ethernet1/18

interface Ethernet1/19
interface Ethernet1/20
  no shutdown

interface Ethernet1/21
  ip address 6.2.0.1/24
  mpls ip forwarding
  no shutdown

interface Ethernet1/21.1
  encapsulation dot1q 1211
  vrf member evpn
  ip address 6.22.0.1/24
  no shutdown

interface Ethernet1/21.2
  encapsulation dot1q 1212
  ip address 6.222.0.1/24
  no shutdown

interface Ethernet1/21.3
  encapsulation dot1q 1213
  vrf member ch5_swap
  ip address 16.1.1.1/24
  no shutdown

interface Ethernet1/22
  no shutdown

interface Ethernet1/23
  description underlay
  ip address 6.1.0.1/24
  mpls ip forwarding
  no shutdown

interface Ethernet1/23.1
  encapsulation dot1q 1231
  vrf member evpn
  ip address 6.11.0.1/23
  no shutdown

interface Ethernet1/24
  no shutdown

interface Ethernet1/25
  no shutdown

interface Ethernet1/26
  description underlay
  ip address 6.0.0.1/24
  mpls ip forwarding
  no shutdown

interface Ethernet1/26.1
  encapsulation dot1q 1261
  ip address 7.0.0.1/24
  no shutdown

interface Ethernet1/27
  no shutdown

interface Ethernet1/28
  no shutdown

interface Ethernet1/29
no shutdown
interface Ethernet1/30
  no shutdown
interface Ethernet1/31
  ip address 1.31.1.1/24
  no shutdown
interface Ethernet1/32
  no shutdown
interface Ethernet1/33
  ip address 87.87.87.1/24
  ip router ospf 100 area 0.0.0.0
  no shutdown
interface Ethernet1/34
  no shutdown
interface Ethernet1/35
  no shutdown
interface Ethernet1/36
  no shutdown
interface mgmt0
  vrf member management
  ip address 172.31.145.107/21
interface loopback1
  ip address 58.58.58.58/32
interface loopback6
  description used for SR underlay testing
  ip address 6.6.6.1/32
line console
line vty
  monitor session 1
    source interface Ethernet1/21 rx
    source interface Ethernet1/23 both
    destination interface sup-eth0
mpls static configuration
  address-family ipv4 unicast
    lsp SL_AGG_BELL
      in-label 6001 allocate policy 88.1.1.0 255.255.255.0
      forward
      path 1 next-hop 6.0.0.2 out-label-stack implicit-null
router ospf 100
  redistribute direct route-map ALL
router bgp 600
  address-family ipv4 unicast
    network 6.6.6.1/32 route-map SRmap
    network 66.1.1.0/24 route-map ULAY_NETWORK
    redistribute direct route-map passall
    maximum-paths 32
    allocate-label all
neighbor 6.0.0.2
  remote-as 50
  ebgp-multihop 255
  address-family ipv4 labeled-unicast
neighbor 6.1.0.2
  remote-as 50
ebgp-multihop 255
address-family ipv4 labeled-unicast
neighbor 6.6.6.3
  remote-as 300
  update-source loopback6
ebgp-multihop 255
address-family vpnv4 unicast
send-community
send-community extended
next-hop-self
import l2vpn evpn reoriginate
neighbor 7.0.0.2
  remote-as 50
ebgp-multihop 255
address-family ipv4 labeled-unicast
neighbor 21.21.21.21
  remote-as 600
  update-source loopback1
address-family l2vpn evpn
send-community
send-community extended
import vpn unicast reoriginate
vrf evpn
address-family ipv4 unicast
  advertise l2vpn evpn
  redistribute direct route-map passall
  redistribute hmm route-map passall
address-family ipv6 unicast
  redistribute direct route-map passall